Ultimate Fighting Championship Fighter Analysis

Spencer Eanes

May 20, 2019

```
fighter.df <- readRDS("~/ADM/fighters.RDS")
```

Scraping the data

I pulled this data from ufcstats.com (https://ufcstats.com) using rvest and the chrome SelectorGadget extension. See fighter_scrape.R for the code. The data set only contains fighters that have fought in the UFC, which is generally considered the premier mixed martial arts promotion in the world. The data was scraped on Sunday May 5th, and thus does not contain any changes that have been posted after that point (the last event that was in the dataset was Fight Night 151 Cowboy vs Iaquinta).

Columns

I collected 24 columns of information - all thirteen stats formally listed on the ufcstats fighter pages, as well as their wins and losses from their record, total fights, win ratio, the number of fights they fought in the UFC, ufc wins, losses, and win ratio, and finally number of UFC wins by Knockout or Technical Knockout, number by submission, and number by Decision. See any fighter page (http://ufcstats.com/fighter-details/029eaff01e6bb8f0) for abbreviated column heading descriptions. The data initially included 3256 fighters, however that is trimmed down to 1180 when removing incomplete cases. For all of my analysis this will be further trimmed down to 793 fighters who have 5 or more fights in the UFC. One small problem is that I am not able to tell if a fighter is male or female based on the data (though if they are in a weight class of 155lb or higher they are male, and if they are in the 115lb weight class they are female - shared weight classes are 125,35,45).

Exploratory Data Analysis

```
dim(fighter.df)
```

```
## [1] 1180   24
```

```
ufc.df <- fighter.df[fighter.df$nUfcFights>=5,]
nrow(ufc.df)
```

```
## [1] 793
```

```
weight_classes <- ufc.df %>%
  group_by(weight)%>%
  summarise(count= length(weight))

weight_classes
```

```
## # A tibble: 9 x 2
##   weight count
##    <dbl> <int>
## 1    115    22
## 2    125    46
## 3    135    93
## 4    145    81
## 5    155   151
## 6    170   150
## 7    185   109
## 8    205    78
## 9    265    63
```

Interestingly, the 155lb and 170lb weight classes are vastly larger than any others. I would attribute this to two causes: men of average height are most likely to fall in this weight range, and fighers will fight in the lowest weight class they can to try to have a size advantage on their opponent.

As somewhat of an afterthought to this analysis (I've done this after pretty much everything else is done), I realized it might be interesting to record reach advantage compared to weight class average rather than pure reach. Another way to approach this could be reach minus height (also known as ape index). However, I prefer to compare reach to weight class average.

```
average.reach <- ufc.df %>%
  group_by(weight) %>%
  summarize(weight.reach = mean(reach))

average.height <- ufc.df %>%
  group_by(weight) %>%
  summarize(weight.reach = mean(height))

average.reach
```

```
## # A tibble: 9 x 2
##   weight weight.reach
##    <dbl>        <dbl>
## 1    115         63.9
## 2    125         66.4
## 3    135         68.3
## 4    145         70.2
## 5    155         71.2
## 6    170         73.2
## 7    185         74.7
## 8    205         75.9
## 9    265         77.4
```

```
average.height
```

```
## # A tibble: 9 x 2
##   weight weight.reach
##    <dbl>        <dbl>
## 1    115         63.6
## 2    125         65.8
## 3    135         66.8
## 4    145         68.6
## 5    155         69.5
## 6    170         71.3
## 7    185         72.8
## 8    205         73.7
## 9    265         74.9
```

```
avg.reach <- data.matrix(average.reach)
rownames(avg.reach) <- t(average.reach[,1])

indiv.reach <- apply(data.matrix(ufc.df$weight),1,function(x) avg.reach[which(x==avg.reach[,1]),
2])
head(indiv.reach)
```

```
## [1] 77.36508 74.74312 63.90909 74.74312 73.22000 68.27957
```

```
ufc.df$reach.diff <- ufc.df$reach-indiv.reach
```

As expected, the most dominant fighter of all time, Jon Jones, has the longest reach differential, 8.1 inches!!!

```
ufc.df$name[which.max(ufc.df$reach.diff)]
```

```
## [1] "Jon Jones"
```

```
ufc.df$reach[which.max(ufc.df$reach.diff)]
```

```
## [1] 84
```

```
ufc.df$reach.diff[which.max(ufc.df$reach.diff)]
```

```
## [1] 8.102564
```

winningness by reach a large reach advantage?

```
large.reach <- ufc.df$reach.diff > 2
summary(lm(overallWinRatio ~ large.reach ,data=ufc.df))
```

```
##
## Call:
## lm(formula = overallWinRatio ~ large.reach, data = ufc.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36633 -0.06059 -0.00204  0.06145  0.28367
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.716330   0.003706 193.306   <2e-16 ***
## large.reachTRUE 0.011775   0.008436   1.396    0.163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09375 on 791 degrees of freedom
## Multiple R-squared:  0.002457,   Adjusted R-squared:  0.001196
## F-statistic: 1.948 on 1 and 791 DF,  p-value: 0.1632
```

Not really…

PCA and K-Means Clustering

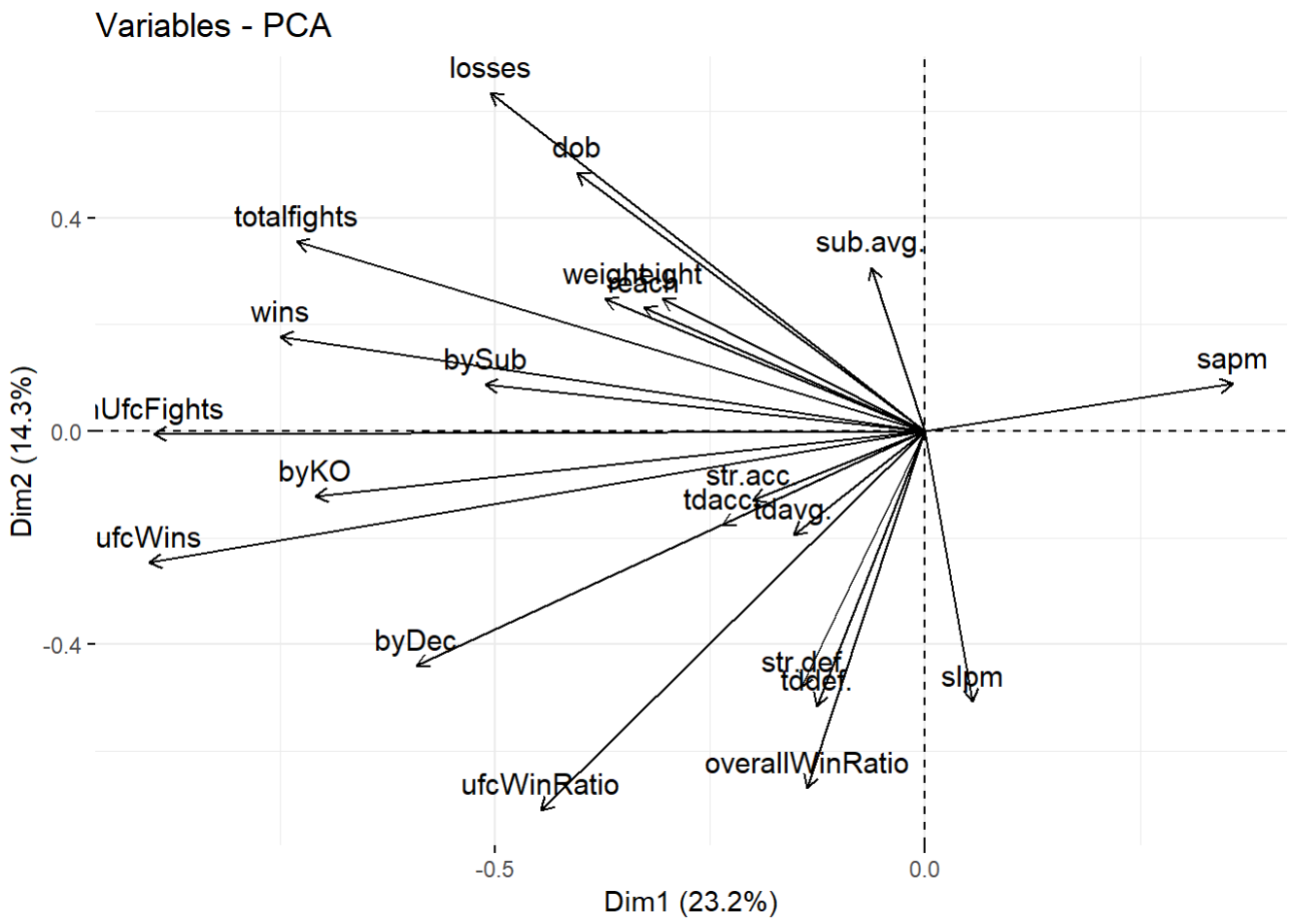
Let's go ahead and do PCA with all our numeric responses and see why this is not ideal. First we'll remove non-numeric columns and scale the data.

```
names(ufc.df)
```

```
##  [1] "name"            "wins"            "losses"
##  [4] "totalfights"     "overallWinRatio" "height"
##  [7] "weight"          "reach"           "stance"
## [10] "dob"             "slpm"            "str.acc."
## [13] "sapm"            "str.def"         "tdavg."
## [16] "tdacc."          "tddef."          "sub.avg."
## [19] "nUfcFights"      "ufcWins"         "ufcWinRatio"
## [22] "byKO"            "bySub"           "byDec"
```

```
ufc.df1 <- ufc.df[,c(-1,-9)]
ufc.df1 <- data.frame(scale(ufc.df1))
rownames(ufc.df1) <- ufc.df$name
```

```
ufc.pca <- prcomp(ufc.df1)
fviz_pca_var(ufc.pca)
```



So what's wrong here? Well, a lot of the constructed variables are non-unique, that is they share some information with other columns. For instance number of UFC wins is included in total wins, which is included in total fights, which is included in overallWinRatio. Another example is that height, reach, and weight are all closely related. So, lets trim down the columns to avoid this. I will remove wins and losses, overallWinRatio, weight, height & reach,, and make byKO, bySub and byDec percentages relative to total ufc wins. I will also remove nUfcFights, ufcWins, and ufcWinRatio. Hopefuly in doing so we can get groupings that are true to the fighters fighting style rather than influenced by winningness, or weight class.

```
names(ufc.df)
```

```
##  [1] "name"           "wins"           "losses"
##  [4] "totalfights"    "overallWinRatio" "height"
##  [7] "weight"         "reach"          "stance"
## [10] "dob"            "slpm"           "str.acc."
## [13] "sapm"           "str.def"        "tdavg."
## [16] "tdacc."         "tddef."         "sub.avg."
## [19] "nUfcFights"     "ufcWins"        "ufcWinRatio"
## [22] "byKO"           "bySub"          "byDec"
## [25] "reach.diff"
```

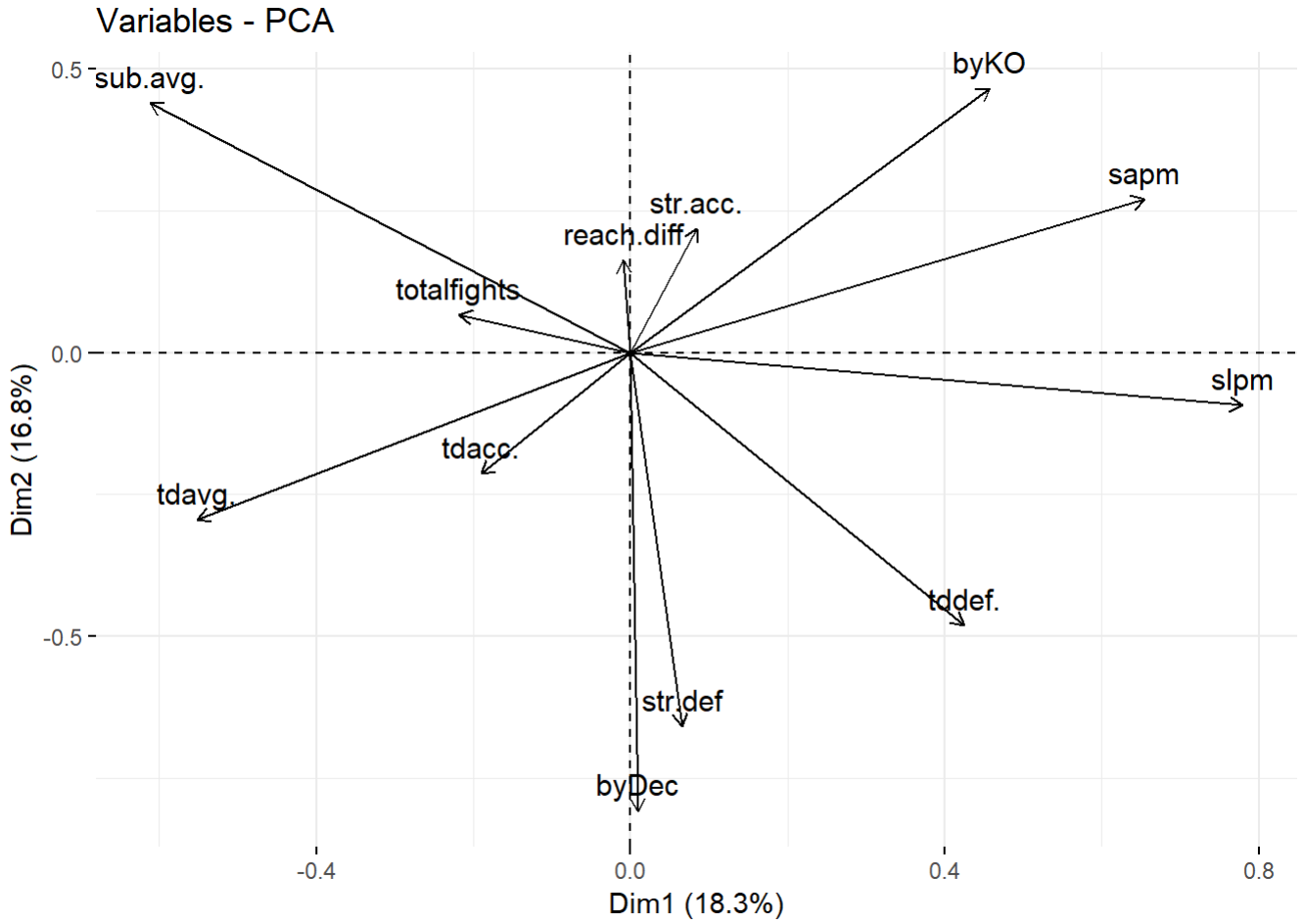
```
ufc.df2 <- ufc.df[,c(-1,-4,-5,-7:-9)]
ufc.df2$byKO <- ufc.df2$byKO / ufc.df2$ufcWins
ufc.df2$bySub <- ufc.df2$bySub / ufc.df2$ufcWins
ufc.df2$byDec <- ufc.df2$byDec / ufc.df2$ufcWins
ufc.df2 <- ufc.df2[,c(-13:-15)]
ufc.df2$totalfights <- ufc.df2$wins+ufc.df2$losses
ufc.df2 <- ufc.df2[c(-1,-2,-4,-14)]

names(ufc.df2)
```

```
##  [1] "height"      "slpm"        "str.acc."    "sapm"        "str.def"
##  [6] "tdavg."      "tdacc."      "tddef."      "sub.avg."    "byKO"
## [11] "byDec"       "reach.diff"  "totalfights"
```

```
ufc.df2 <- ufc.df2[,-1]
ufc.df2 <- data.frame(scale(ufc.df2))
rownames(ufc.df2) <- ufc.df$name
```

Now let's do PCA on this reduced dataset with less overlapping data.

```
ufc.pca <- prcomp(ufc.df2)
fviz_pca_var(ufc.pca)
```



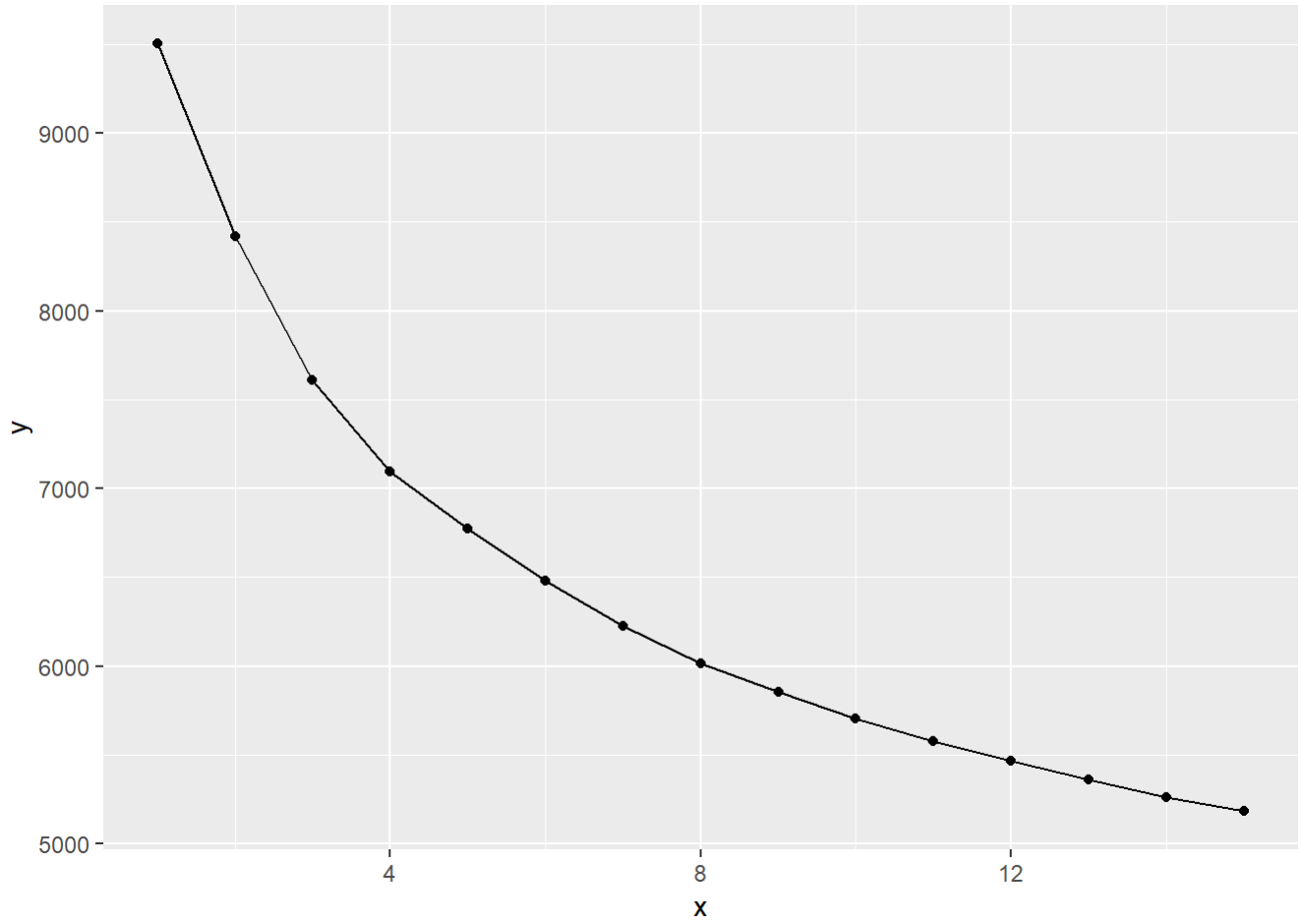
As a domain expert, the two directions I would pull out of this plot follow the axes in this case (that's nice!). This is a little bit hard to tell because some of the variables contribute in two directions. This will be explained below. The x-axis represents preferred method of fighting, with stand-up fighters on the right, represented bystrikes landed and absorbed, wins by KO, take down defence in this direction. In the negative x-direction is ground fighters, represented by wins by submission and takedown. On the y-axis is how fighters win, by finish (a finish is KO/TKO or submission) in the positive y-direction, represented again by submission average, percent wins by KO, strike accuracy, and interestingly reach differential (which does make sense). In the negative y-direction are fighters than win by decision prominently, and those with good take down and strike defence, as well as those with many takedowns.

```
rots <- ufc.pca$rotation
ufc.pca1<- as.matrix(ufc.df2) %*% rots
ufc.pca1 <- data.frame(ufc.pca1)
dim(ufc.pca1)
```

```
## [1] 793  12
```

```
pca1 <- ufc.pca1[,1]
pca2 <- ufc.pca1[,2]
pca3 <- ufc.pca1[,3]
```

```
k.max <- 15
wss <- sapply(1:k.max, function(x){ kmeans(ufc.df2,x,nstart=30,iter.max=20)$tot.withinss } )
data.frame(x=1:k.max,y=wss)%>%
  ggplot(aes(x=x,y=y))+geom_point()+geom_line()
```




```
K <- 4
```

In my opinion the optimal number of groups looks to be between 4 and 8. We will go with 4 groups to be conservative.

```
ufc.km <- kmeans(ufc.df2,K,iter.max=20,nstart=30)
ufc.km1 <- kmeans(ufc.pca1,K,iter.max=20,nstart=30)
table(ufc.km$cluster,ufc.km1$cluster)
```

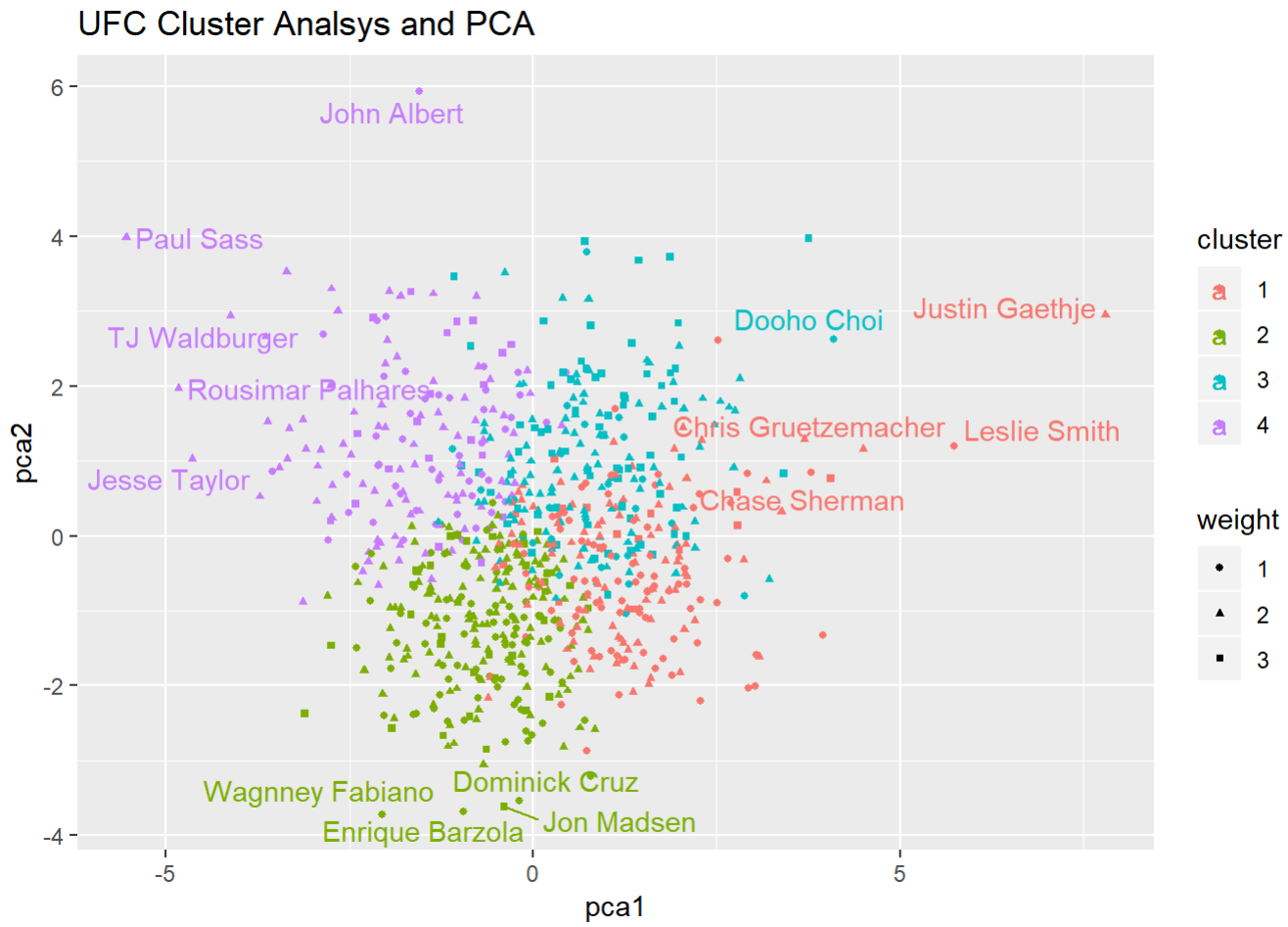
```
##
##       1   2   3   4
##   1   0   0   0 195
##   2   0   0 244   0
##   3 197   0   0   0
##   4   0 157   0   0
```

With and without PCA the two clusterings are in total agreement. Perfect!

```
ufc.df2$cluster.km<- ufc.km$cluster
ufcCluster.df <- data.frame(pca1,pca2,cluster=factor(ufc.df2$cluster.km),name=ufc.df$name,weight
=as.factor(ifelse(ufc.df$weight<155,1,ifelse(ufc.df$weight<205,2,3))))
```

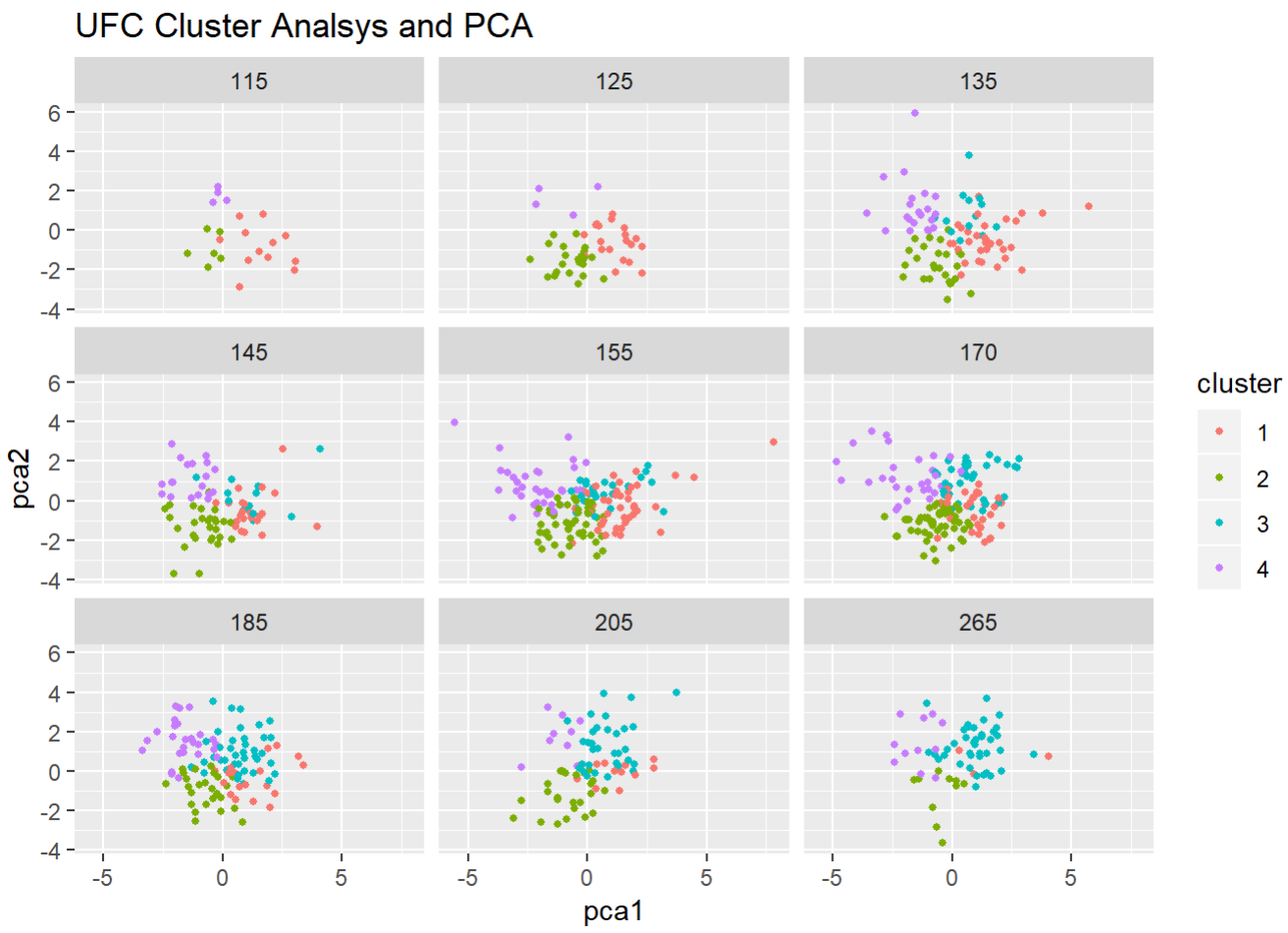
Here is a plot of all fighters based on these directions, with stand-outs names marked.

```
gp2 <- ggplot(ufcCluster.df,aes(pca1,pca2,color=cluster,shape=weight))+
  geom_point(size=1)+
  #geom_text_repel(aes(label=name),size=3)+
  geom_text_repel(data = subset(ufcCluster.df, pca1 < -4), aes(label = name))+
  geom_text_repel(data = subset(ufcCluster.df, pca1 > 4), aes(label = name))+
  geom_text_repel(data = subset(ufcCluster.df, pca2 > 4.25), aes(label = name))+
  geom_text_repel(data = subset(ufcCluster.df, pca2 < -3.5), aes(label = name))+
  ggtitle("UFC Cluster Analsys and PCA")

gp2
```



Let's facet wrap by weight and see if we really did achieve a model that is invariant of weight classes.

```
ufcCluster.df$weight <- ufc.df$weight

ggplot(ufcCluster.df,aes(pca1,pca2,color=cluster))+
  geom_point(size=1)+
  facet_wrap(~ weight, ncol=3)+
  ggtitle("UFC Cluster Analsys and PCA")
```



Looks like there is still some bias between weight classes, but perhaps this is to be expected. There tend to be more knockout punchers in heavier weight classes which makes sense because weight tends to transfer well to one punch knockout power in MMA, with Derrick “The Black Beast” Lewis being a very good example of this.

Doing the K-means cluster analysis using PCA, we can see that the data relatively splits along the four quadrants, where, in order of quadrants we have (1) hard-hitting knockout artists, (2) grapplers and submission artists, (3) wrestlers who dominate on top for decision victories and (4) evasive strikers who go to decision. Let’s plot this with top 10 fighters and other notables in each division to see if this theory continues to hold.

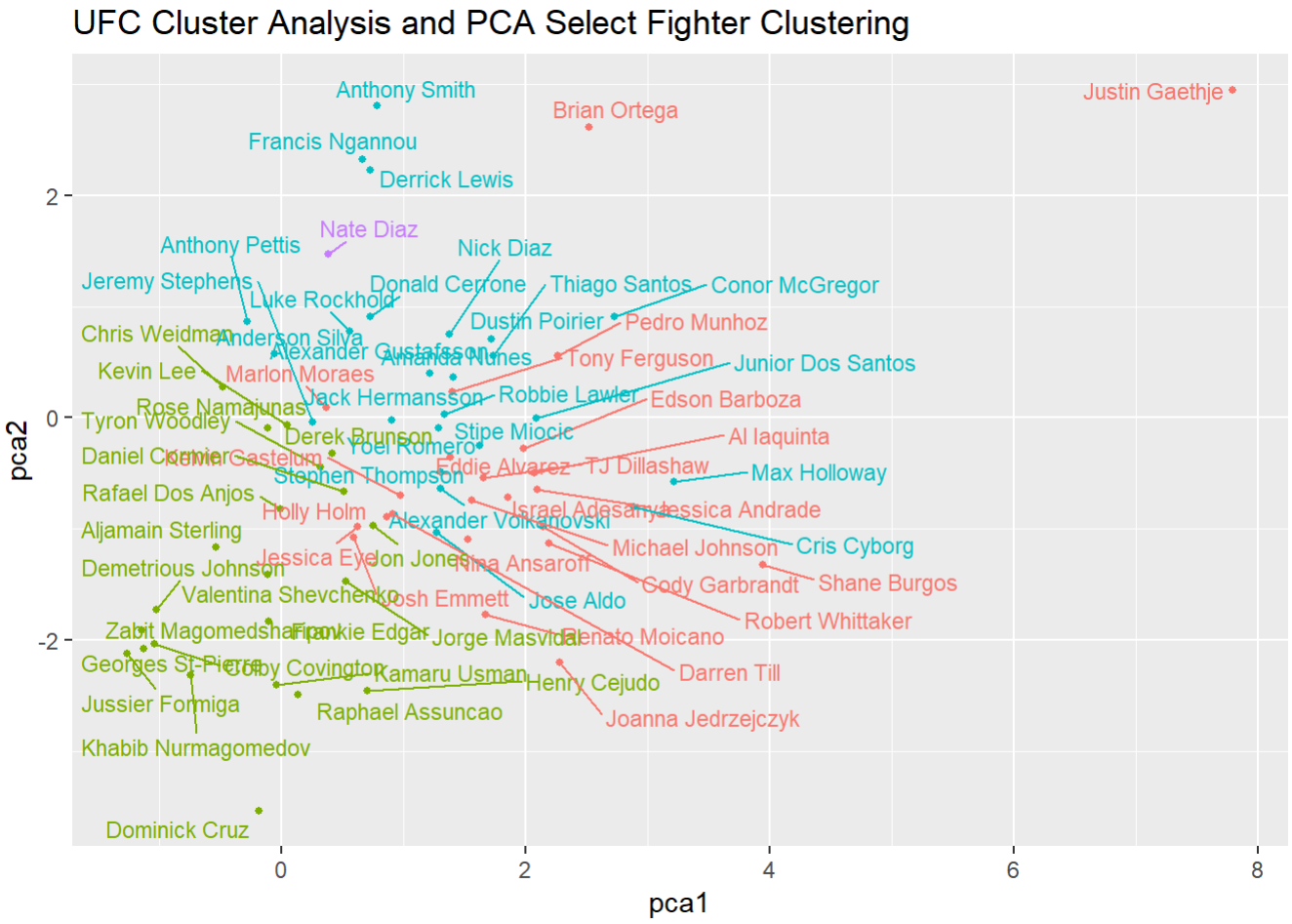
```
notable_fighters <- read.csv("~/ADM/notable_fighters.txt")
notable_fighters <- apply(notable_fighters,2,as.character)

nf.df <- ufc.df2[notable_fighters,]

indices <- as.numeric(lapply(notable_fighters, function(n) which(ufc.df$name==n))) %>% na.omit

nf.df <- nf.df[complete.cases(nf.df),]
nrow(nf.df)
```

```
## [1] 70
```

The fighters whose names are in the notable fighters documents are more well known recent fighters (particularly who I am familiar with so that I can interpret this more accurately). They are all currently or formerly ranked in the top 10 of their weight division. This set includes all division champions and most top 5 with a few others thrown in.

Below let's try reclustering this group on it's own, and seeing how this compares to how they were clustered with all the others.

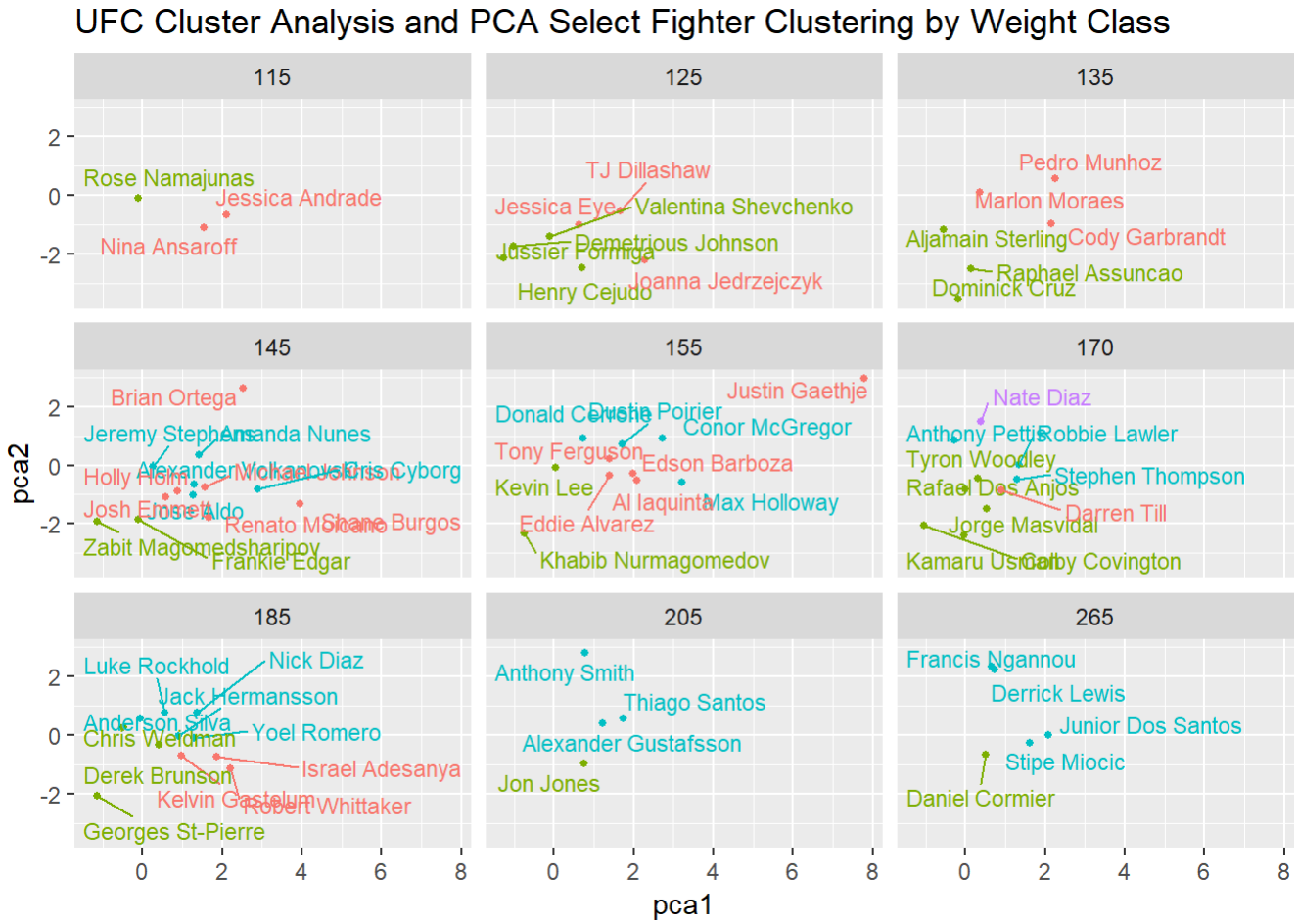
```
nf.df$pca1 <- pca1[indices]
nf.df$pca2 <- pca2[indices]

nf.df$name <- ufc.df$name[indices]
weights <- ufc.df[indices,]$weight
nf.df$weight <- weights

gp1 <- ggplot(nf.df,aes(pca1,pca2,color=as.factor(cluster.km)))+
  geom_point(size=1)+
  geom_text_repel(aes(label=name),size=3)+
  guides(color=F)

gp1 + ggtitle("UFC Cluster Analysis and PCA Select Fighter Clustering")
```



Gaethje is the real stand-out here, something that I actually expected before I even finished parsing the data. This is because his style is so unique - he walks forward head down gaurd up and attempts to leg kick opponents until they can't continue, disregarding his own health. He has *by far* the highest strikes landed and absorbed per minute in the UFC. Perhaps he could be called the Harden of the UFC, though he is known to some by his fan given nickname, Justin "CTE" Gaethje. The one other thing to note is that Nate Diaz is the only grappler in this group. I would say at higher MMA levels you can't be only a grappler, but need to be more well rounded in other areas, which probably leads to fewer for the notable fighters group. Additionally, at a high level almost everyone has a lot of grappling experience, which means submissions are less common.

Let’s facet wrap this by weight class.

```
levels(nf.df$weight) <- c("115","125","135","145","155","170","185","205","265")


gp1 +
  facet_wrap(~ weight, ncol=3, labeller=label_value)+
  ggtitle("UFC Cluster Analysis and PCA Select Fighter Clustering by Weight Class")
```



This really works out even better than I could have hoped for. Fighters styles really match with the group they are placed into. I will mention a few notables.

* Zabit Magomedsharipov (145lbs) and Khabib Nurmagomenov (155lbs) - two Dagestani cousins who are known for their sambo and chain wrestling “smesh” style can both be found far down in the lower left.
* Usman and Covington of the 170lbs weight class, both known for a similar American grind wrestling style are also found in the lower left.
* Max Holloway (155lbs in this plot due to his most recent fight, but the 145lb champion) can be found on the right middle of the plot, a volume striker with good takedown defence who has a fair number of TKO’s and decisions.
* Jon Jones (205lbs), one of the most dominant fighters in history can be found fairly centrally, as he is a very well-rounded fighter.
* Brian Ortega (145lbs) can be found between strikers and grapplers despite being known for his submission game. I would guess this is because of his most recent fight with aforementioned champion Max Holloway in a four round war that left Ortega with the most strikes ever absorbed in a single fight. This skews his SAPM up, and places him more in the striking realm despite being a grappler.
* The two olmpyic wrestlers, Daniel Cormier (265lbs) and Henry Cejudo (125lbs) can both be found in the wrestling group.

Hierarchical Clustering

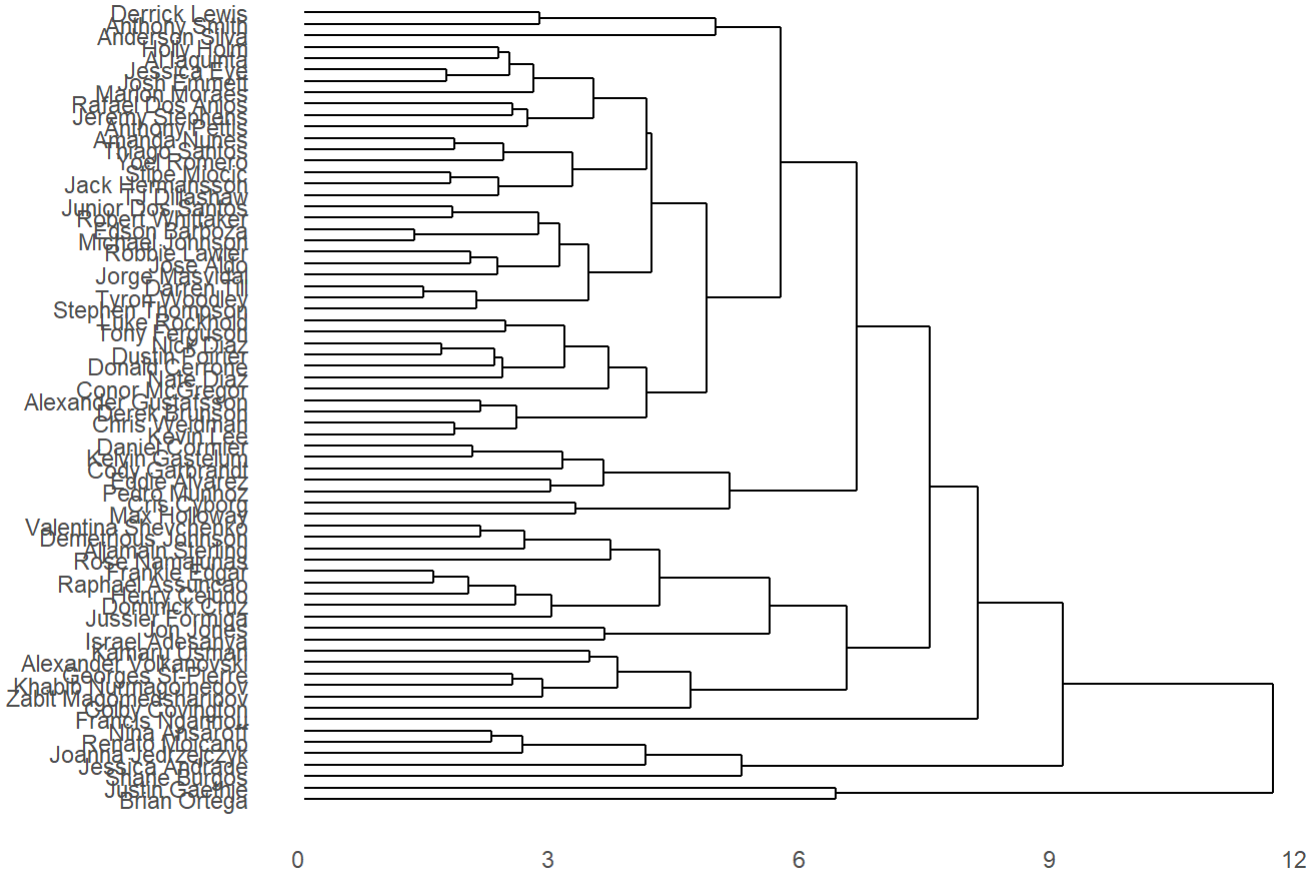
Let's do some hierachical clustering and see where the pieces fall. I would recommend running these plots on their own and making them bigger for viewing.

```
names(nf.df)
```

```
##  [1] "slpm"        "str.acc."    "sapm"        "str.def"     "tdavg."
##  [6] "tdacc."      "tddef."      "sub.avg."    "byKO"        "byDec"
## [11] "reach.diff"  "totalfights" "cluster.km"  "pca1"        "pca2"
## [16] "name"        "weight"
```

```
nf.df <- nf.df[,-17:-13]
nf.dist <- dist(nf.df)
```

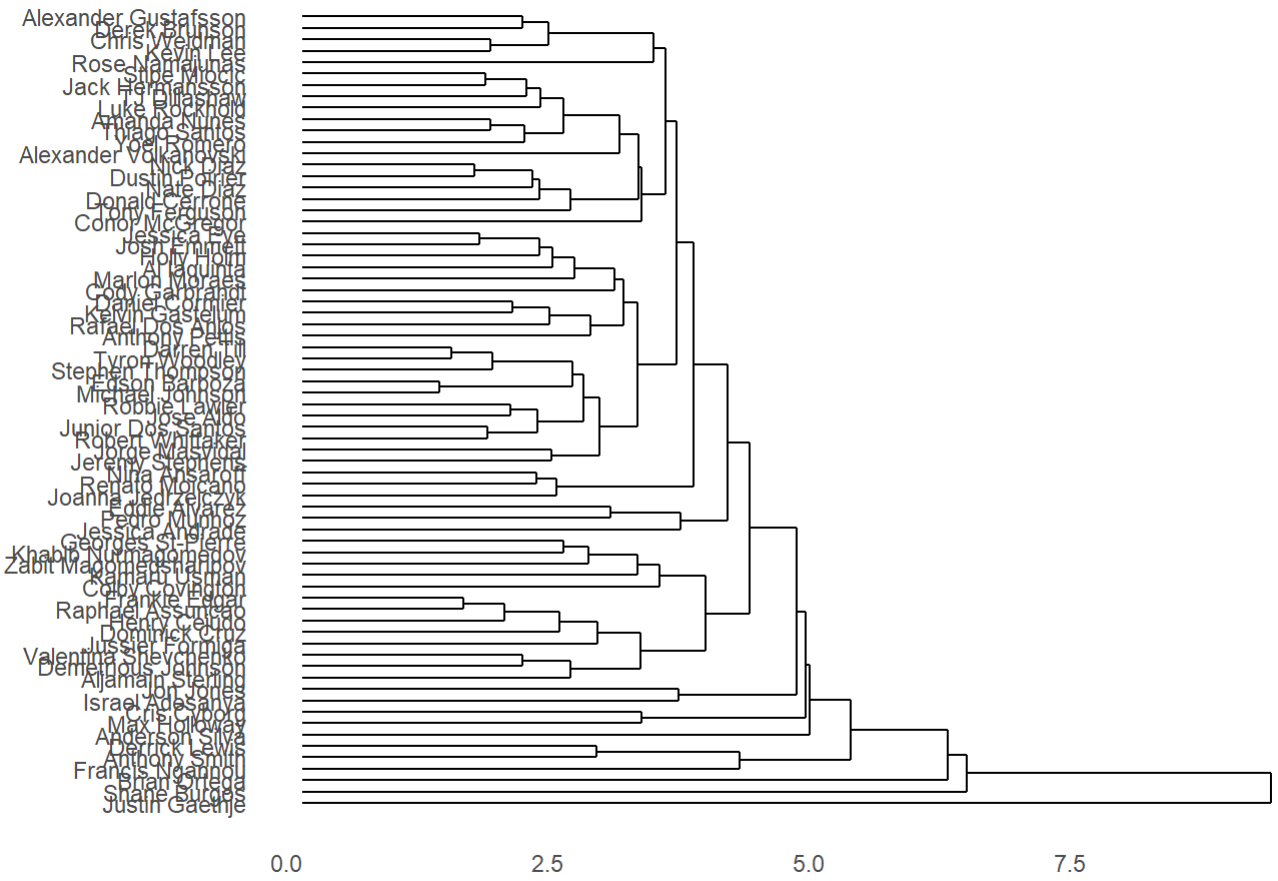
```
nf.hc.c <- hclust(nf.dist,method="complete")
par(mfrow=c(1,1))

ggdendrogram(nf.hc.c,rotate=T)
```



I like the complete grouping quite a lot. Most of the smaller groups make a lot of sense Anthony Smith and Derrick Lewis are both heavy power punchers who will knock you down or out. Iaquinta and Holm are both strong boxers who tend to take fights into later rounds. Daniel Cormier and Kelvin Gastelum is an interesting comparison I hadn't

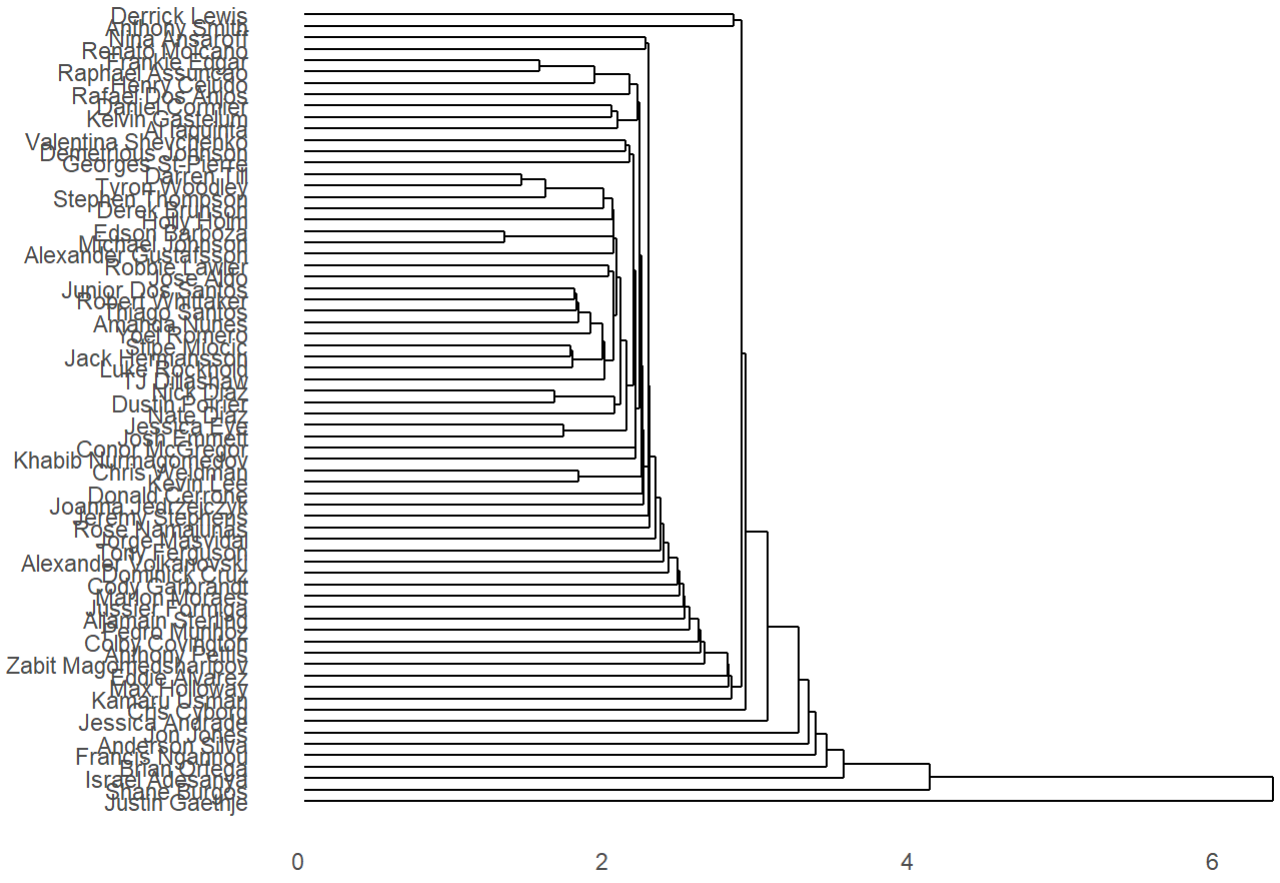
considered before but makes sense - both ahve a good wrestling pedigree but tend to look for knockouts on the feet before going to the ground. Jones and Adesanya is also an easy comparison to see for most fans. Finally, Khabib and Zabit are grouped together as above. I'm not really a fan of Darren Till and Tyron Woodley being so close together, but I'm guessing Woodley may be slightly misclassified despite being more of a knockout puncher due to a few recent full length bouts.

```
nf.hc.c <- hclust(nf.dist,method="average")
par(mfrow=c(1,1))

ggdendrogram(nf.hc.c,rotate=T)
```

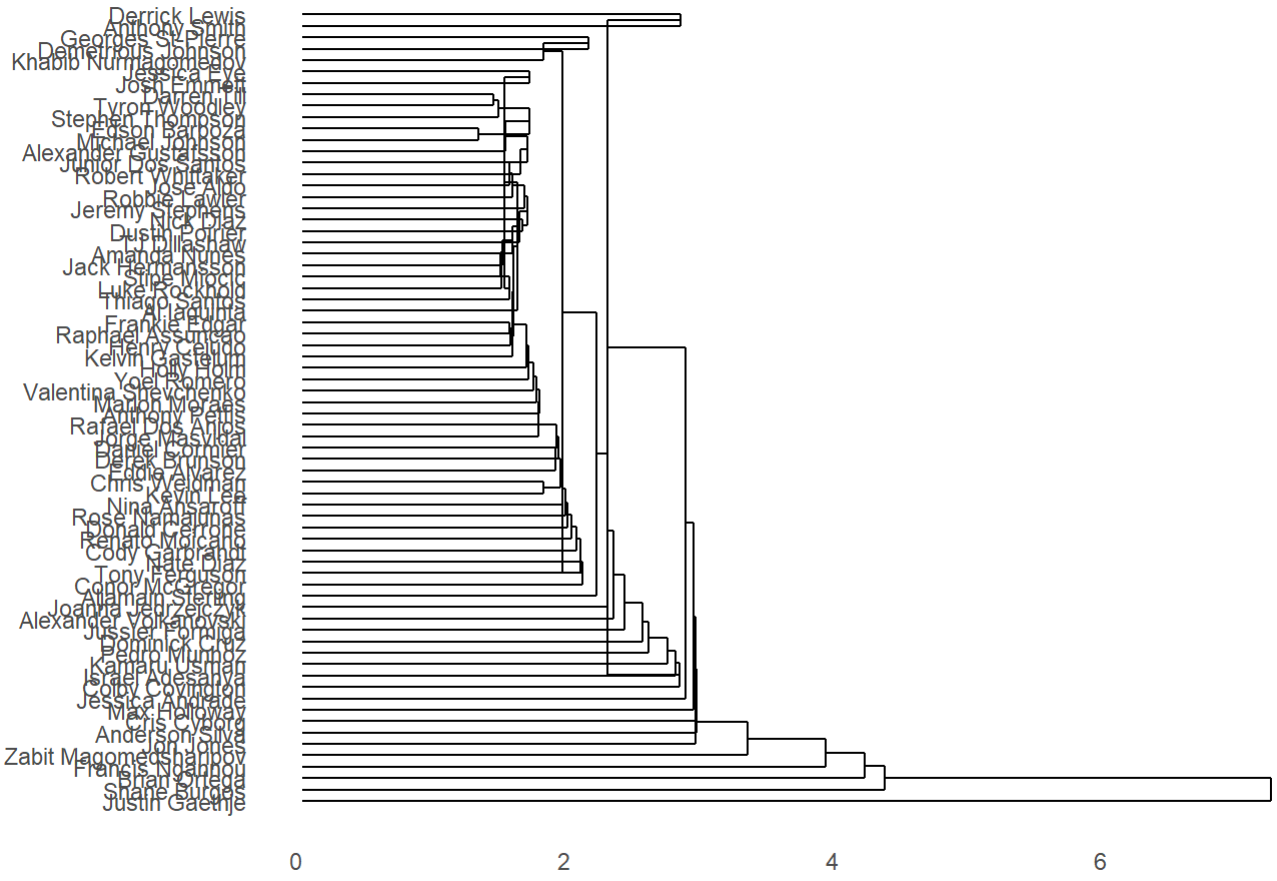


I'm not as big of a fan of this clustering. I don't like the Gustafsson and Brunson comparison - the first being a boxer and the second a wrestler, nor the Alvarez - Jedrzejczyk comparison, the first being more of a brawler and the second a light finesse striker. There are a lot of similar comarisons here as to complete, but a few changes I don't agree with.

```
nf.hc.c <- hclust(nf.dist,method="single")
par(mfrow=c(1,1))

ggdendrogram(nf.hc.c,rotate=T)
```




```
nf.hc.c <- hclust(nf.dist,method="centroid")
par(mfrow=c(1,1))

ggdendrogram(nf.hc.c,rotate=T)
```



I think that both of single and centroid clustering also leave much to be desired. The more pairwise comparisons that join later don't give the same nice stylistic groupings like the previous two methods. Overall I personally prefer the complete method clustering.

Predicting UFC Win Ratio

I am also interested in the predictive power of fighter stats and their UFC performance. If the UFC win and loss information is removed from the data frame (as below in ufc.df3), this is in my opinion a reasonable question to ask. For the most part, I would expect a fighters non-ufc record to be from before they joined the UFC. If a fighter were to leave and the UFC and fight in another promotion, I am somewhat doubtful whether or not their new wins and losses would be included this data set since it is taken from UFCstats.com. Let's strip the dataset down and do some prediction.

```
names(ufc.df)
```

```
##  [1] "name"            "wins"            "losses"
##  [4] "totalfights"     "overallWinRatio" "height"
##  [7] "weight"          "reach"           "stance"
## [10] "dob"             "slpm"            "str.acc."
## [13] "sapm"            "str.def"         "tdavg."
## [16] "tdacc."          "tddef."          "sub.avg."
## [19] "nUfcFights"      "ufcWins"         "ufcWinRatio"
## [22] "byKO"            "bySub"           "byDec"
## [25] "reach.diff"
```

```
ufc.df$wins <- ufc.df$wins-ufc.df$ufcWins
ufc.df$losses <- (ufc.df$nUfcFights - ufc.df$ufcWins)

ufc.df3 <- ufc.df[c(-1,-4,-5,-24:-22,-20,-19)]
names(ufc.df3)
```

```
##  [1] "wins"        "losses"      "height"      "weight"      "reach"
##  [6] "stance"      "dob"         "slpm"        "str.acc."    "sapm"
## [11] "str.def"     "tdavg."      "tdacc."      "tddef."      "sub.avg."
## [16] "ufcWinRatio" "reach.diff"
```

Sample out a training and testing set.

```
samp <- sample(1:nrow(ufc.df3),nrow(ufc.df3)/2,rep=F)
train.df <- ufc.df3[samp,]
test.df <- ufc.df3[-samp,]
```

Let’s chuck it all in a linear model and see what we get.

```
ufcwin.lm <- lm(ufcWinRatio ~ ., data=train.df)
summary(ufcwin.lm)
```

```
##
## Call:
## lm(formula = ufcWinRatio ~ ., data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38477 -0.08518  0.01153  0.09149  0.34717
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.1242939  0.3155218   0.394 0.693855
## wins              -0.0019599  0.0009453  -2.073 0.038815 *
## losses            -0.0108480  0.0028733  -3.775 0.000186 ***
## height            -0.0060323  0.0050912  -1.185 0.236821
## weight             0.0002391  0.0004768   0.501 0.616337
## reach              0.0080631  0.0067268   1.199 0.231420
## stanceopenstance  -0.0096184  0.1055399  -0.091 0.927433
## stanceorthodox     0.0158103  0.0454504   0.348 0.728141
## stancesouthpaw     0.0111550  0.0479067   0.233 0.816005
## stanceswitch      -0.0349735  0.0601613  -0.581 0.561366
## dob               -0.0023005  0.0017956  -1.281 0.200906
## slpm               0.0635372  0.0096403   6.591 1.48e-10 ***
## str.acc.           0.0500741  0.1133888   0.442 0.659022
## sapm              -0.0607133  0.0092568  -6.559 1.80e-10 ***
## str.def            0.4022192  0.1229121   3.272 0.001165 **
## tdavg.             0.0066132  0.0067362   0.982 0.326858
## tdacc.            -0.0079085  0.0374569  -0.211 0.832896
## tddef.             0.1098465  0.0432145   2.542 0.011426 *
## sub.avg.           0.0433023  0.0111849   3.871 0.000128 ***
## reach.diff        -0.0029276  0.0064672  -0.453 0.651042
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1311 on 376 degrees of freedom
## Multiple R-squared:  0.4082, Adjusted R-squared:  0.3783
## F-statistic: 13.65 on 19 and 376 DF,  p-value: < 2.2e-16
```

```
uw.p.lm <- predict(ufcwin.lm,newdata=test.df)
mean((uw.p.lm-test.df$ufcWinRatio)^2)
```

```
## [1] 0.01999052
```

This model gives some really impressive performance, an error rate of about 2%, especially considering this is a response from 0 to 1 (essentially logistic) modeled by a linear model. However, I believe this is probably due to one trait of the table - the majority of the actual fight statistics are derived only from UFC fights. This means that we are assuming that these numbers are representative of the non-UFC record, which is not necessarily the case.

Also notice that strikes landed and absorbed per minute (SLPM and SAPM) have some of the highest predictive power (along with previous win and sub.avg.). This makes a lot of sense - fighters who hit more than they are hit win more and vice versa. The fact that SAPM and SLPM are recorded from UFC fights is probably not very fair.

These both probably depend more on winning or losing than they predict them for the average fighter. This could be different from other stats, for instance strike accuracy could be consistent across promotions. Let's remove them from the data, re-run and see what we get.

```
ufc.df3 <- ufc.df3[,c(-8,-10)]
names(ufc.df3)
```

```
##  [1] "wins"        "losses"      "height"      "weight"      "reach"
##  [6] "stance"      "dob"         "str.acc."    "str.def"     "tdavg."
## [11] "tdacc."      "tddef."      "sub.avg."    "ufcWinRatio" "reach.diff"
```

```
K <- 5
group <- sample(1:K,nrow(ufc.df3),rep=T)
lmcv <- numeric(K)
for(i in 1:K){

  train.df <- ufc.df3[group!=i,]
  test.df <- ufc.df3[group==i,]
  ufcwin.lm <- lm(ufcWinRatio ~ ., data=train.df)
  summary(ufcwin.lm)
  uw.p.lm <- predict(ufcwin.lm,newdata=test.df)
  #if it's outside 0-100%, move it to the correct end
  uw.p.lm[uw.p.lm < 0] <- 0
  uw.p.lm[uw.p.lm > 1] <- 1
  lmcv[i] <- mean((uw.p.lm-test.df$ufcWinRatio)^2)
}

(lm.mse <- mean(lmcv))
```

```
## [1] 0.02045908
```

In the absence of SAPM and SLPM other UFC stats become more predictive, takedown defence, submission average, strike accuracy and defence in particular. So really this didn't do much. Personally, from a qualitiative standpoint, I think it makes most sense to keep SAPM and SLPM out of the data set, and to hope the other stats are relatively representative.

Ridge regression

It makes much more sense to do this regression with logistic regression since the response is a percentage between 0 and 1. Glmnet has a really nice way to do this, where I can provide a two column reponse of losses and wins and it internally expands this to replicate the observation into a binary response with the correct numbers of wins and losses represented. Let's try out some penalized regression and see how this compares to linear regression.

```
suppressMessages(library(glmnet))

ufclosses <- ufc.df$nUfcFights-ufc.df$ufcWins
outcome <- data.matrix(cbind(ufclosses,ufc.df[,"ufcWins"]))
colnames(outcome) <- c("losses","wins")
rownames(outcome) <- ufc.df$name
head(outcome)
```

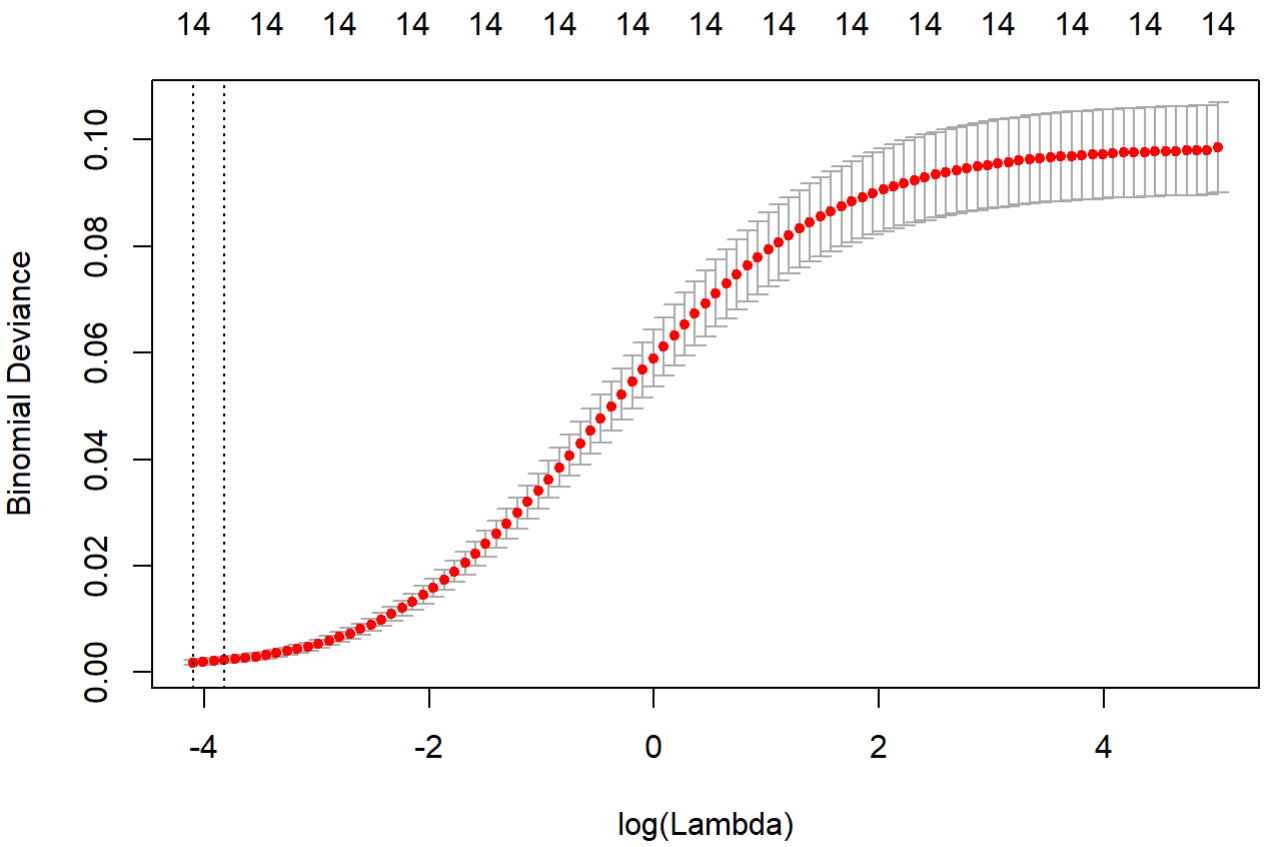
```
##                     losses wins
## Shamil Abdurakhimov      2    5
## Israel Adesanya          0    6
## Jessica Aguilar          4    1
## Omari Akhmedov           4    6
## Yoshihiro Akiyama        8   14
## John Albert              4    1
```

```
predictors <- data.matrix(ufc.df3[,-1*ncol(ufc.df3)])
head(predictors)
```

```
##    wins losses height weight reach stance dob str.acc. str.def tdavg.
## 4    15      2     75    265    76      3  38     0.44    0.60   1.34
## 13   11      0     76    185    80      5  30     0.51    0.65   0.00
## 18   19      4     63    115    63      3  37     0.50    0.53   0.94
## 22   12      4     72    185    73      3  32     0.34    0.57   2.51
## 23    0      8     70    170    73      3  44     0.41    0.57   2.29
## 28    6      4     68    135    68      1  33     0.49    0.35   0.00
##    tdacc. tddef. sub.avg. ufcWinRatio
## 4    0.24   0.66      0.2   0.7142857
## 13   0.00   0.85      0.5   1.0000000
## 18   0.25   0.50      0.2   0.2000000
## 22   0.48   0.58      0.3   0.6000000
## 23   0.67   0.92      1.0   0.6363636
## 28   0.00   0.50      6.1   0.2000000
```

```
samp <- sample(1:nrow(outcome),nrow(outcome)/2,rep=F)
train.x <- predictors[samp,]
test.x <- predictors[-samp,]

train.y <- outcome[samp,]
test.y <- ufc.df3[-samp,"ufcWinRatio"]
```

```
#I'll trust glmnet decide to the lambda values
ufc.ridge <- cv.glmnet(train.x,train.y,
                    family="binomial",
                    alpha=0)

plot(ufc.ridge)
```




```
(best.lambda <- ufc.ridge$lambda.min)
```

```
## [1] 0.01655891
```

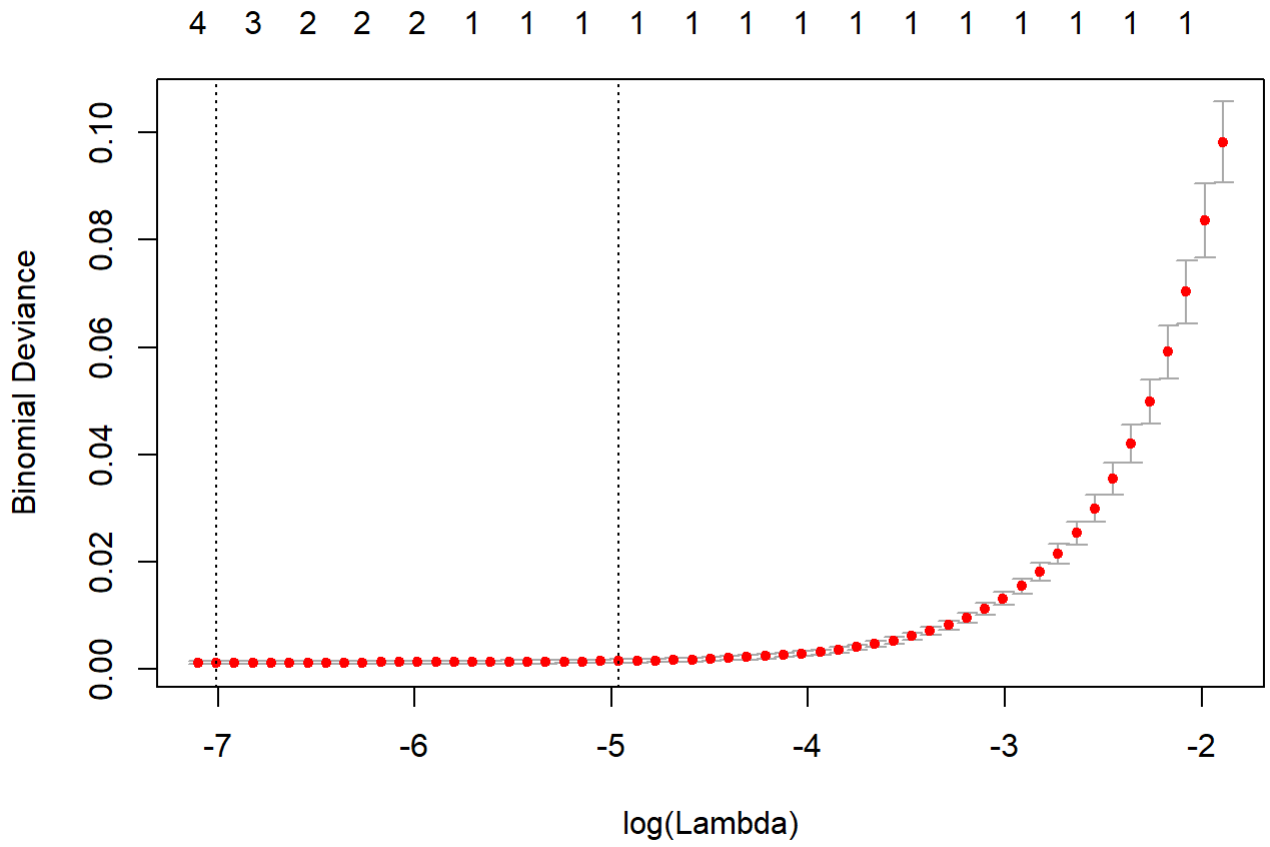
```
pred <- predict(ufc.ridge,newx=test.x,s=best.lambda,type="response")
head(pred)
```

```
##            1
## 4  0.7174765
## 18 0.2293565
## 22 0.6053876
## 23 0.6444817
## 40 0.8182614
## 41 0.6038589
```

```
(ridge.mse <- mean((test.y-pred)^2))
```

```
## [1] 0.0004047355
```

```
ufc.lasso <- cv.glmnet(train.x,train.y,
                       family="binomial",
                       alpha=1)

plot(ufc.lasso)
```




```
(best.lambda <- ufc.lasso$lambda.min)
```

```
## [1] 0.0009044936
```

```
pred <- predict(ufc.lasso,newx=test.x,s=best.lambda,type="response")
head(pred)
```

```
##            1
## 4  0.7212718
## 18 0.2099087
## 22 0.6086845
## 23 0.6450197
## 40 0.8293613
## 41 0.6094907
```

```
(lasso.mse <- mean((test.y-pred)^2))
```

```
## [1] 0.0002428545
```

```
c(lm.mse,ridge.mse,lasso.mse)
```

```
## [1] 0.0204590751 0.0004047355 0.0002428545
```

Wow - penalized logistic regression does a fantastic job of predicting win rates. If I could get a dataset that contained these same stats averaged over a fighters whole career rather than just UFC career I think this would be a better analysis, but alas I must make do with this. This is a decent baseline, and would make a good set up to re do this analysis with the correct information. Based on this analysis it seems like we can *potentially* predict fighters win rates in the UFC assuming that their stats in the UFC are representative of their stats outside the UFC.